

### Amendments to the Claims

Kindly amend claims 1, 3, 4 & 8, and cancel claims 5 & 7 (without prejudice), as set forth below. All pending claims are reproduced below, with changes in the amended claims shown by underlining (for added matter) and strikethrough/double brackets (for deleted matter).

1. (Currently Amended) A non-disruptive method for replacing a first software module of a system with a second software module, the method comprising:

copying, by a currently executing, first software module of a system, update control code from the currently executing, first software module to memory space outside a memory location at which the first software module is currently executing, wherein only the update control code is copied from the currently executing, first software module to the memory space outside the memory location at which the first software module is currently executing;

replacing the currently executing, first software module with a second software module by storing the second software module in memory at a location which at least partially overlies the memory location at which the first software module was executing, wherein the replacing includes employing the update control code copied from the first software module to memory space outside the memory location from which the first software module was executing, and wherein the replacing includes executing the update control code copied from the first software module during the replacing of the first software module with the second software module to monitor replacing of the first software module with the second software module; [[and]]

responsive to the replacing, and without resetting or restarting the system, branching from executing the update control code to executing code of the second software module at an entry point thereof specified by the update control code, thereby beginning execution of the second software module; and without resetting the system

wherein the first software module and the second software module are each a single statically linked module.

2. (Previously Canceled).

3. (Currently Amended) The method of claim 1, wherein the first software module comprises [[a]] an operating firmware module, and the system comprises an embedded system, and wherein the replacing includes overlaying the memory location of the operating firmware module with the second software module, the second software module comprising an updated firmware module.

4. (Currently Amended) The method of claim ~~[[1]]~~ 3, wherein the update control code ~~includes update control code for monitoring replacing of the first software module with the second software module~~ is only a portion of the currently executing, first software module.

5. (Canceled).

6. (Original) The method of claim 1, wherein the first software module includes at least one of a loader and a linker, and wherein the replacing includes overlaying the memory location of the first software module with the second software module.

7. (Canceled).

8. (Currently amended) The method of claim 1, further comprising ~~storing~~ leaving data to be maintained during the replacing of the first software module ~~[[to]]~~ in memory space outside the memory location of the first software module.

9. (Original) The method of claim 1, wherein the system comprises an embedded system, and the first software module and the second software module each comprise a firmware module.

10. (Original) The method of claim 1, wherein the replacing employs a hardware based direct memory access (DMA) operation to save the second software module to a target memory space and wherein the copying update control code comprises copying the update control code to memory space outside the target memory space, and wherein the update control code includes control code for determining when the DMA operation has completed and for branching to an entry point of the second software module upon completion of the DMA operation.

11-31. (Previously Canceled).

\* \* \* \* \*